

- 1 -

## STORAGE SYSTEM

### INCORPORATION BY REFERENCE

The present application claims priority from Japanese application JP2004-037596 filed on February 16, 2004, the content of which is hereby incorporated 5 by reference into this application.

### BACKGROUND OF THE INVENTION

The present invention relates to a storage system for use by a computing system.

JP-A-9-297699 (pages 3 - 4 and Fig. 1) discloses a system referred to as a hierarchical storage system which comprises a computer, and a high speed storage and a low speed storage connected to the computer. In JP-A-9-297699 (pages 3 - 4 and Fig. 1), more frequently used files are stored in a high speed storage such as a magnetic disk drive, while less frequently used files are stored in an inexpensive low speed storage such as a tape drive. Then, a table is used to manage the access frequency for each file, and is referenced to determine which file is allocated to 15 or stored in which storage. 20

### SUMMARY OF THE INVENTION

JP-A-9-297699 relies on software running on the computer to implement a hierarchical storage

control for moving a file between a small capacity magnetic disk drive and a large capacity magnetic tape drive in accordance with how frequently the file is used. The hierarchical storage control assumes that

5 data which has been highly frequently accessed in the past will be highly frequently accessed as well in the future, and determines a storage for storing data therein in accordance with statistic information on the access frequency of the data and an available capacity

10 of a fast accessible storage. Then, the hierarchical storage control improves the processing efficiency and practically manages a large amount of files by increasing the probability that highly frequently accessed data is stored in a fast accessible storage.

15

The conventional hierarchical storage technique, however, has a problem in that the capacity scalability is limited by the number of magnetic disk drives and the number of magnetic tape drives which can be connected to the computer, thereby failing to fully

20 provide long-term management for a large amount of information which increases more and more over time.

It is therefore an object of the present invention to provide a storage system having a capacity scalability which permits a large amount of file

25 information to be managed over a long term.

The storage system of the present invention is configured to have the ability to control input/output to/from an external storage system

connected to the storage system. The storage system of the present invention builds a file system over a storage area locally provided thereby and a storage area provided by the external storage system.

5           The storage system can build NAS which has a capacity scalability.

Other objects, features and advantages of the invention will become apparent from the following description of the embodiments of the invention taken  
10         in conjunction with the accompanying drawings.

#### BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram illustrating an exemplary configuration of a computing system to which the present invention is applied;

15         Fig. 2 is a plan view illustrating an appearance of an exemplary storage;

Fig. 3 is a perspective view illustrating an appearance of an exemplary adaptor board;

20         Fig. 4 is a block diagram illustrating an exemplary configuration of a NAS channel adaptor;

Fig. 5 is a block diagram illustrating exemplary programs stored in a file system control memory;

25         Fig. 6 is a block diagram illustrating exemplary programs stored in a disk array control memory;

Fig. 7 is a block diagram illustrating an

exemplary configuration of a heterogeneous storage connection control channel adaptor;

Fig. 8 is a block diagram illustrating exemplary programs stored in a heterogeneous storage  
5 connection control memory;

Fig. 9 is a block diagram illustrating an exemplary configuration of a heterogeneous storage;

Fig. 10 is a block diagram illustrating an exemplary configuration in which one heterogeneous  
10 storage is connected to each storage;

Figs. 11A to 11C show exemplary structures for a volume management table;

Fig. 12 is a block diagram illustrating an exemplary configuration in which a plurality of  
15 heterogeneous storages are connected to each storage;

Fig. 13 is a block diagram illustrating an exemplary configuration for utilizing a mixture of storages and heterogeneous storages;

Fig. 14 is a block diagram illustrating another exemplary configuration for utilizing a mixture  
20 of storages and heterogeneous storages; and

Figs. 15A to 15C show exemplary structures for the volume management table.

#### DESCRIPTION OF THE EMBODIMENTS

25 In the following, embodiments of the present invention will be described with reference to the accompanying drawings. The present invention, however,

is not limited to the embodiments described below.

To begin with, a first embodiment will be described.

(1) Exemplary System Configuration

5 Fig. 1 illustrates an exemplary computing system in the first embodiment. In the following description, "x" represents an arbitrary integer.

A storage 1 represents a disk array system which has a disk controller 11 (hereinafter called the 10 "DKC 11"); a plurality of magnetic disk drives 17xx (hereinafter called the "disks 17xx"); and a management device 18. Assume in the first embodiment that each disk 17xx is an FC (Fibre Channel) disk which includes a Fibre Channel interface.

15 Next, the configuration of the DKC 11 in the storage 1 will be described for purposes of illustration. The DKC 11 comprises one or a plurality of NAS channel adaptors 110x (hereinafter called the "CHN 110x"); one or a plurality of heterogeneous 20 storage connection adaptors 111x (hereinafter called the "CHD 111x"); a plurality of disk adaptors 12x (hereinafter called the "DKA 12x"); a shared memory 13 (hereinafter called the "SM 13"); a shared memory controller 15 (hereinafter called the "SMC 15"); a 25 cache memory 14 (hereinafter called the "CM 14"); and a cache memory controller 16 (hereinafter called the "CMC 16").

Each CHN 110x is an interface controller

connected to an associated computer 40x (hereinafter called the "host 40x") connected to a local area network 20 (hereinafter called the "LAN 20") through a file I/O interface.

5           Each CHD 111x is an interface controller connected to an associated remote storage 50x (hereinafter called the "heterogeneous storage 50x") connected to a storage area network 30 (hereinafter called the "SAN 30") through a block I/O interface. In  
10          the following description, the CHN and CHD are collectively called a "channel adaptor" (or abbreviated as "CH").

The SMC 15 is connected to the CHN 110x, CHD 111x, DKA 12x, and SM 13. The SMC 15 controls data  
15          transferred among the CHN 110x, CHD 111x, DKA 12x, and SM 13.

The CMC 16 is connected to the CHN 110x, CHD 111X, DKA 12x, and CM 14. The CMC 16 controls data transferred among the CHN 110x, CHD 11x, DKA 12x, and  
20          CM 14.

The SM 13 has a volume management table 131. The volume management table 131 stores the configuration of a "logical device" (hereinafter called the "LDEV") for management. The LDEV constitutes a  
25          logical configuration unit of an internal storage area which comprises a series of logical sequential address spaces.

Each disk 17xx is connected to associated DKA

12x. Each DKA 12x controls input/output to/from one or a plurality of disks 17xx connected thereto.

In the storage 1, every CH can access the CM 14, SM 13, any of the DKA's 12x, and any of the disks 5 17x through the CMC 16 or SMC 15.

The management device 18 is connected to the DKC 11 within the storage 1 for managing the configuration of the storage 1 through each CH and each DKA. Configuration information is stored in the SM 13, 10 and is shared by the respective CH's and DKA's.

The heterogeneous storage 50x is a storage installed external to the storage 1, and is of a type different from the storage 1. The heterogeneous storage 50x is connected to an associated CHD 11x 15 through the SAN 30. When viewed from the heterogeneous storage 50x, the storage 1 is in position of a host computer which issues I/O. It should be noted that while the heterogeneous storage 50x is defined as a type of storage different from the storage 1 in the 20 following description, the heterogeneous storage 50x may be of the same type as the storage 1 in an alternative embodiment.

The LAN 20 connects the CHN's 110x with the associated hosts 40x. Generally, an IP network is used 25 for the LAN.

The SAN 30 connects the CHD's 111x with the associated heterogeneous storages 50x. Generally, Fibre Channel (FC) is used for the SAN. Alternatively,

iSCSI may be used for the SAN in which case an SCSI command conforming to the SCSI protocol is encapsulated in an IP packet for communication among devices connected to the SAN through an IP network. Assume in  
5 the first embodiment that the SAN 30 is provided exclusively for connection to the heterogeneous storages 50x, and is not therefore connected to the hosts 40x.

A management terminal 600 is connected to the  
10 management device 18 included in the storage 1 through a management LAN 70. The management terminal 600 is also connected to the heterogeneous storages 50x through the management LAN 70. The management terminal 600 runs a management software application for setting  
15 and managing the storage 1 and heterogeneous storages 50x.

The storage 1 illustrated in Fig. 1 only has a NAS interface (CHN 110x) for connection with the hosts 40x through the LAN 20. The computing system in  
20 the first embodiment may additionally comprise a SAN interface (SAN channel adaptor) for connecting the storage 1 to the hosts 40x through the SAN 30, such that either the NAS interface or the SAN interface can be selected.

25 (2) Exemplary Appearance of Storage

Fig. 2 illustrates an exemplary appearance of the storage 1.

A DKC unit 19 stores the CHN's 110x, CHD's 11x, DKA's 12x, SM 13, and CM 14 which are components of the DKC 11. The SM 13 is actually comprised of a plurality of controller boards 13x. Likewise, the CM 5 14 is comprised of a plurality of cache boards 14x. The user of the storage 1 can increase or decrease the number of these boards to tailor the storage 1 that has a desired storage capacity of the CM 14 or SM 13. A disk unit (hereinafter called the "DKU") 180 and DKU 10 181 store a plurality of disks 17xx.

Each of slots 190 receives an adaptor board which contains the CHN's 110x, CHD's 111x, DKA's 12x, controller boards 13x, cache boards 14x, and the like. In the first embodiment, the shape of the slot 190, the 15 size of the adaptor board, and the shape of a connector are consistent irrespective of the type of adaptor board and the type of interface, so that the compatibility is ensured. Consequently, an arbitrary adaptor board can be plugged into an arbitrary slot 190 20 of the DKC unit 19 irrespective of the type of adaptor board or the type of interface. Also, the user of the storage 1 can freely select the number of adaptor boards for the CHN's 110x and CHD's 111x to plug the selected number of CHN's 110x and CHD 111x into slots 25 190 of the DKC unit 19.

(3) Exemplary Appearance of CHN Board

Fig. 3 illustrates an exemplary appearance of

the adaptor board (hereinafter called the "CHN board") which contains the CHN 110x. A connector 11007 is connected to a connector of the DKC unit 19. An interface connector 2001 can be connected to the LAN 5 20.

In the first embodiment, since the connector of the adaptor board is consistent in shape irrespective of the type of adaptor board, the CHN board has a connector in the same shape as an adaptor 10 board (hereinafter called the "CHD board") which contains the CHD 111x. It should be noted that the interface connector 2201 of the CHD board supports the fibre channel, and is designed for connection with the fibre channel.

15 (4) Exemplary Configuration of NAS Channel Adaptor  
(CHN)

Fig. 4 illustrates an exemplary configuration of the CHN 110x. A file access control CPU 11001 is a processor for controlling accesses to files. A LAN 20 controller 11002 is connected to the LAN 20 through the interface connector 2001 for controlling transmission/reception of data to/from the LAN 20. A file access control memory 11004 is connected to the file access control CPU 11001. The file access control 25 memory 11004 stores programs executed by the file access control CPU 11001, and associated control data.

A disk array control CPU 11008 is a processor

for controlling a disk array. The disk array used herein has groups, each of which is composed of a plurality of disks 17xx and is handled as a single virtual disk, and operates the plurality of disks 17xx

5 in parallel to improve the performance. Particularly, a so-called RAID (Redundant Arrays of Inexpensive Disks) is a disk array which stores redundant data called "parities" in part of a storage area of a group to increase the fault tolerance. Among disk arrays,

10 the RAID is particularly frequently used.

A disk array control memory 11009, which is connected to the disk array control CPU 11008, stores programs executed by the disk array control CPU 11009, and associated control data. An SM I/F control circuit

15 11005 controls an access from the CHN 110x to the SM

13. A CM I/F control circuit 11006 controls an access from the CHN 110x to the CM 14. An inter-CPU communication circuit 11007 is used when the file access control CPU 11001 communicates with the disk

20 array control CPU 11008 for accessing a disk.

While the first embodiment illustrates an example of an asymmetric multi-processor configuration in which the CHN 110x is mounted with two processors, i.e., the file access control CPU 11001 and disk array

25 control CPU 11008, the CHN may be mounted with a single processor which executes both of the file access control and disk array control. Further alternatively, the CHN 110x may be designed in a symmetric multi-

processor configuration which employs two or more processors that evenly execute one or both of the file access control and disk array control.

(5) Exemplary Configuration of File Access Control

5       Memory

Fig. 5 illustrates, by way of example, programs and associated control data stored in the file access control memory 11004 included in the CHN 110x. An operating system program 110040 is used to manage 10 whichever programs associated therewith and control input/output operations. A LAN controller driver program 110041 is used to control the LAN controller 11002. A TCP/IP program 110042 is used to control TCP/IP which is a communication protocol on the LAN. A 15 network file system program 110044 is used to control NFS, CIFS, and the like which are protocols for providing the NAS host 40x with files stored in the storage. A volume control program 110045 is used to control a logical volume comprised of one or a 20 plurality of logical units (hereinafter called the "LU"). An inter-CPU communication driver program 110046 is used to control the inter-CPU control circuit 11007 for communicating between the file access control CPU 11001 and disk array control CPU 11008.

25           A file system program 110043 is used to manage files stored in the storage, and executes file storage management and input/output control.

Specifically, the file system program 110043 is involved in control processing including:

- 1) opening a file when it is to be used;
- 2) responding to a file access request
- 5 received from a host to execute disk input/output operations in accordance with the access request;
- 3) determining an area on a disk in which a file is stored for management; and
- 4) managing a correspondence relationship
- 10 between the name of an opened file and a table which manages a file storage area and a buffer address of the file.

(6) Exemplary Configuration of Disk Array Control Memory

15 Fig. 6 illustrates, by way of example, programs and associated control data stored in the disk array control memory 11009 included in the CHN 110x.

An operating system program 110090 is used to manage whichever programs associated therewith and

20 control input/output operations.

A driver program used for a CPU to communicate with another CPU (hereinafter called an inter-CPU communication driver program) 110093 is used to control the inter-CPU communication circuit 11007

25 for communicating between the file access control CPU 11001 and disk array control CPU 11008, and receives an access request for the LU from the file access control

CPU 11001.

A volume control program 110092 creates one or a plurality of logical devices (hereinafter called the "LDEV"), each of which is a logical configuration 5 unit of a storage area comprising a pair of logical sequential address spaces, on a RAID group (hereinafter called the "VDEV") comprised of a plurality of disks 17xx to form a RAID, couples one or a plurality of LDEV's to create a logical unit (hereinafter called the 10 "LU"), and manages relation information associated with the logical devices and logical unit.

A cache control program 110094 is used for management of data stored in the CM 14, and for control such as determination of cache hit/miss.

15 A DKA communication driver program 110095 is used to communicate with the DKA 12x when a disk 17xx must be accessed.

A disk array control program 110091 is involved in a sequence of disk array control 20 operations. Specifically, from an access request from the file access control CPU 11001 to the LU received through the inter-CPU communication driver program 110093, the disk array control program 110091 identifies an LDEV and a VDEV corresponding to the LU 25 accessed by the file access control CPU 11001 with the aid of the volume control program 110092, determines cache miss or hit associated with the access with the aid of the cache control program 110094, and issues an

access request to the DKA 12x with the aid of the DKA communication driver program 110095 when an access to a disk is required.

(7) Exemplary Configuration of Heterogeneous Storage

5 Connection Adaptor (CHD)

Fig. 7 illustrates an exemplary configuration of the CHD 111x.

A heterogeneous storage connection control CPU 11108 is a processor for controlling a connection 10 to a heterogeneous storage 50x.

A heterogeneous storage connection control memory 11109, which is connected to the heterogeneous storage connection control CPU 11108, stores programs executed by the heterogeneous storage connection CPU 15 11108, and associated control data. An SM I/F control circuit 11105 controls an access from the CHD 110x to the SM 13. A CM I/F control circuit 11106 controls an access from the CHD 110x to the CM 14.

(8) Exemplary Configuration of Heterogeneous Storage

20 Connection Control Memory

Fig. 8 illustrates, by way of example, programs and associated control data stored in the heterogeneous storage connection control memory 11109 included in the CHD 111x.

25 An operating system program 111090 is used to manage whichever programs associated therewith and

control input/output operations.

A heterogeneous storage connection control program 111091 recognizes a heterogeneous storage 50x connected to the SAN 30, confirms the capacity of an LU 5 provided by the heterogeneous storage 50x, and reads from and writes into the LU.

A volume control program 111092 regards an LU provided by the heterogeneous storage 50x as one of VDEV's contained in the storage 1 for management. 10 Since this LU is handled as a VDEV within the storage 1, an LDEV is created on the VDEV. When the volume control program 111092 manages an LU in the heterogeneous storage 50x as a VDEV, this VDEV is corresponded to the LDEV by the volume control program 15 110092 of the CHN 110x, and the LDEV in turn is corresponded to the LU.

It should be noted that the LU in the heterogeneous storage 50x is handled as a VDEV, but when the heterogeneous storage unit 50x is a RAID disk 20 array device, no redundant data need be added to the storage 1 because the heterogeneous storage 50x comprises a RAID within itself.

A cache control program 111093 is used for management of data stored in the CM 14 and for control 25 such as determination of cache hit/miss.

An LDEV migration control program 111094 is involved in a migration of the LDEV.

When the data stored in an LDEV managed by

the storage 1 is copied to an LDEV created on the heterogeneous storage 50x and the original data is erased, the LDEV appears as if it has moved to the heterogeneous storage 50x. This sequence of operations  
5 is called the "LDEV migration." The LDEV migration control program 111094 executes an LDEV migration between the storage 1 and heterogeneous storage 50x; an LDEV migration within the heterogeneous storage 50x; and an LDEV migration between one heterogeneous storage  
10 50x and another heterogeneous storage 50x.

A WORM control program 111095 adds a WORM (Write Once Read Many) attribute to a LDEV created in a heterogeneous storage 50x managed by the CHD 1110. An example of WORM control may involve aborting all write  
15 operations other than a write operation associated with a migration of an LDEV, and handling this LDEV as a read only LDEV to give the WORM attribute to the LDEV. Otherwise, the WORM control may permit a write-once operation to an LDEV to give the LDEV the WORM  
20 attribute. The following description will be made on the assumption that all write operations are aborted except for a write operation associated with a migration of an LDEV, and the LDEV is handled as a read only LDEV.

25 (9) Exemplary Configuration of Heterogeneous Storage

Fig. 9 illustrates an exemplary configuration of the heterogeneous storage 50x. The heterogeneous

storage 50x comprises a pair of disk array controllers FCTLx510x for providing fault tolerance, such that when one FCTL becomes inoperative due to a fault, the other is substituted for the faulty one to take over.

5           Each of the disk array controllers FCTLx510x comprises an FC controller 51012, a disk array control CPU 51008, a disk array control memory 51009, a cache memory 510144, a data transfer control circuit 51011, and an FC controller 51010. The FC controller 51012  
10 connects the FCTLx510x to the SAN 30.

The disk array control CPU 51008 is a processor for controlling a disk array. The disk array control memory 51009 stores a disk array control program and associated control data. The FC controller 15 51010 connects and controls the disks 5710x. The data transfer control circuit 51011 is disposed among the FC controller 51012, CM 51014, and FC controller 51010 for controlling data input/output to/from the CM 51014, and a data transfer to the other FCTL.

20           Like the storage 1, the heterogeneous storage 50x forms a plurality of disks 5710x into a RAID group, and creates one or a plurality of LU's, each of which is allocated a sequence of addresses in part or all of its storage area.

25           In Fig. 9, CHD0 (1110) of the storage 1 is connected to a heterogeneous storage 0(500) through a heterogeneous storage connection SAN 30, and operates as if it were a host, when viewed from the

heterogeneous storage 0(500). The CHDO (1110) acts as a host of the heterogeneous storage 0(500) to issue access commands such as a read and a write for controlling the heterogeneous storage 0(500).

5 (10) Exemplary Configuration: Single Heterogeneous Storage Connected to Heterogeneous Storage Connection SAN

In the following, the operation of the storage system according to the first embodiment will 10 be described with reference to the configurations described above.

Fig. 10 illustrates in a simplified form an exemplary configuration of a computing system for highlighting features of the system while omitting the 15 detailed configurations described above.

Figs. 11A - 11C show exemplary states of the volume management table 131 corresponding to Fig. 10.

The volume management table 131 shown in Fig. 11A represents an exemplary relationship between LU's 20 created in the heterogeneous storage 500 and VDEV's made up of LU's created in the heterogeneous storage 500. In the volume management table 131 shown in Fig. 11A, "SLUN" indicates an identification number of the LU created and used in the heterogeneous storage 500, 25 i.e., an identification number used by the disk array controller FCTLx510x in the heterogeneous storage 500 for accessing the LU. "VDEV" indicates a VDEV

corresponding to a LU created in the heterogeneous storage 500.

The volume management table 131 shown in Fig. 11B represents an exemplary relationship between the 5 VDEV shown in Fig. 11A and LDEV. In the volume management table 131 shown in Fig. 11B, "LDEV" indicates an LDEV corresponding to the VDEV.

The volume management table 131 shown in Fig. 11C represents an exemplary relationship among the LDEV 10 shown in Fig. 11B, LUN, LV, and FS. In the volume management table 131 shown in Fig. 11C, "LUN" indicates the number of an LU corresponding to an LDEV; "LV" indicates a number corresponding to the LU; and "FS" indicates an FS corresponding to the LV.

15 In the first embodiment, the volume management table 131 contains information on the FS for management, however, the FS may be managed by a separate management table, so that the management of the FS is not limited to the particular manner employed 20 in this embodiment.

In the storage 1 of Fig. 10, the configuration of the CHN 1110 illustrated in Fig. 4 is generally called a "NAS function 1100A." Also, the configuration of the CHD 1110 illustrated in Fig. 7 is 25 generally called a "heterogeneous storage connection control function 1110A."

The operation of the first embodiment will be described in connection with an example in which a

logical device 5200 (SLDEV0) is created on a LU with SLUN=0 in the heterogeneous storage 500, and a file system 5300 (SFS0) is created on the logical device SLDEV0.

5           Assume that the heterogeneous storage 500 has created the LU with SLUN=0 through the disk array controller FCTLx510x of the heterogeneous storage 500.

Referring first to Fig. 8, description will be made on the operation of the heterogeneous storage  
10 connection control function 1110A of the CHD0 (1110).

A heterogeneous storage connection control program 111091 issues an inquiry command to the heterogenous storage 500 to detect LU0 in the heterogeneous storage 500. A volume control program 111092 regards this LU0  
15 as SVDEVO, and registers the same in the volume management table 131 of the SM 13. Specifically, the volume control program 111092 registers "0" in the SLUN entry and SVDEVO in the VDEV entry in the volume management table 131 shown in Fig. 11A.

20           Referring next to Fig. 6, description will be made on the operation of the NAS function 1100A of the CHNO (1100). The volume control program 110092 executed by the disk array control CPU 11008 in Fig. 6 references the volume management table 131 shown in  
25 Fig. 11A, stored in the SM 13, to detect the SVDEVO. The volume control program 110092 creates SLDEV0 of a proper size for this SVDEVO, and registers the created SLDEV0 in the volume management table 131 shown in Fig.

11B stored in the SM 13. Specifically, the volume control program 110092 registers SVDEV0 in the VDEV entry, and the SLDEV0 corresponding to the SVDEV0 in the LDEV entry of the volume management table 131 shown 5 in Fig. 11B.

The manager issues an instruction from the management terminal 18 to the computing system for creating an LU having a desired capacity, and upon receipt of the instruction, the storage 1 creates the 10 LU comprised of one or a plurality of LDEV's. Assume herein that one LU0 is created with one SLDEV0. In other words, the volume control program 110092 registers SLDEV0 in the LDEV entry and the number "0" of the LU0 corresponding to SLDEV0 in the LUN entry of 15 the volume control program 110092.

Next, as the host 40x activates the storage 1, the volume control program 110045 in Fig. 5 executed by the file access control CPU 11001 in the storage 1 issues an inquiry command to the disk array control CPU 20 11008 using the inter-CPU communication driver program 110046 to make a query to the disk array control CPU 11008 for detecting the LU0. The volume control program 110092 executed by the disk array control CPU 11008 detects the LU0, and notifies the file access 25 control CPU 11001 of the detection. The volume control program 110045 recognizes this LU0, and creates a logical volume LV0 using the LU0. While a logical volume can be created by coupling a plurality of LU's,

assume herein that the LV0 is created with the single LU0. In other words, the volume control program 110045 registers LV0 corresponding to the LU0 in the LV entry of the volume management table 131 shown in Fig. 11C.

5           In response to an instruction from the manager, the file system program 110043 creates a file system SFS0 on the logical volume LV0. In other words, the file system program 110043 registers SFS0 corresponding to the LV0 in the FS entry of the volume 10 management table 131 shown in Fig. 11C.

With the foregoing operations, the LV0 in the storage 1 is created from the LU0 of the storage 1, while the LU0 is created from the SLDEVO in the first embodiment. The SLDEVO in turn includes the storage 15 area SVDEVO implemented by the LU in the heterogeneous storage system. As a result, the logical device SLDEVO controlled by the CHNO (1100) is established on the heterogeneous storage 500, and the file system SFS0 controlled by the CHNO (1100) is established on the 20 SLDEVO.

As the host 0(400) issues a query to the file system, the LAN control driver program 110041 illustrated in Fig. 5 controls the LAN controller 11002 to receive a packet including this query from the LAN 25 20, and the file system program 110043 recognizes this query by the action of the TCP/IP program 110042 and network file system program 110044. Then, the file system program 110043 transmits directory information

on the SFS0 to the host 0(400) which then recognizes that this file system is the SFS0 resident in the storage 1, as illustrated in Fig. 10. Subsequently, the file system can come into use. Here, the directory 5 information on the file system SFS0 is general directory information, so that this directory information is not described in detail in the present disclosure.

In the first embodiment, the disks 17xx contained in the storage 1 are not at all required, while information for managing the file system SFS0 created by the NAS function 1100A is stored in the SLDEV0 established on the LU in the heterogeneous storage 50 through the heterogeneous storage connection 15 SAN 30 by the action of the heterogeneous storage connection control function 1110A of the CHD0. Here, the information for managing the file system SFS0 is general file system management information commonly referred to as metadata, so that detailed description 20 will not be made on its structure in the present disclosure.

As described above, according to the configuration illustrated in Fig. 10, the NAS function 1100A of the storage 1 can create a file system on 25 disks contained in the heterogeneous storage 500 connected to the storage 1 through the heterogeneous storage connection control function 1110A. This function will be hereinafter called the "heterogeneous

storage connection NAS function."

When the host 0(400) is to access a file stored in the file system FS0, the host 0(400) sends an access request to the CHNO (1100) for specifying a file system name and a file name. Upon receipt of the access request including the file system name and file name, the CHNO (1100) references the volume management table 131 to determine the location at which the associated file is stored. When the file is stored in a heterogeneous storage, the CHNO (1100) informs the CHDO (1110) that an access is requested for data stored in the heterogeneous storage, and simultaneously notifies the CHDO (1110) of the data storage location retrieved from the volume management table 131, thus permitting the CHDO (1110) to access data in the heterogeneous storage. Also, the CHNO (1100) may store information on an access request from the host to data in the heterogeneous storage in the SM 13, such that the CHDO (1110) periodically checks the SM 13 for information on an access request from the host to data in the heterogeneous storage. Upon recognition of an access request, the CHDO (1110) may reference the volume management table 131 to access data in the heterogeneous storage. When the CHDO (1110) accesses the heterogeneous storage 500, the CHDO (1110) is required to use addresses which can be recognized by the disk array controller FCTLx510x of the heterogeneous storage 500. Thus, the CHDO (1110)

accesses the heterogeneous storage 500 using the SLUN  
(in the volume management table 131 in Fig. 11A)  
corresponding to the VDEV which stores the data.

5 (11) Exemplary Configuration: Plurality of  
Heterogeneous Storages Connected to Heterogeneous  
Storage Connection SAN

Next described will be made on an exemplary configuration in which a plurality of heterogeneous storages are connected to the heterogeneous storage connection SAN.  
10

connection SAN.

Fig. 12 illustrates an exemplary configuration in which a plurality of heterogeneous storages are connected to the heterogeneous storage connection SAN. The configuration in Fig. 12 differs from the configuration in Fig. 10 in that a plurality of (two in Fig. 12) heterogenous storages 500, 501 are connected to the heterogeneous storage connection SAN  
15  
30.

Operations similar to those performed in the configuration of Fig. 10 are duplicated for the heterogenous storages 500, 501 to create a file system SFS0 on the SLDEVO in the heterogeneous storage 500, and a file system SFS1 on SLDEV1 in the heterogeneous storage 501. The file systems SFS0 and SFF1 are both  
25  
function 1100A of the CHNO.

In this way, by connecting an increased

number of heterogeneous storages 50x to the heterogeneous storage connection SAN 30, it is possible to increase the storage capacity which can be handled by the CHNO.

5 As described above, according to the configuration in Fig. 12, an arbitrary number of heterogeneous storages can be connected to the heterogeneous storage connection SAN 30 of the storage 1, thereby making it possible to build a large scaled  
10 NAS which has a capacity scalability.

As will be appreciated from the foregoing, according to the first embodiment, the ability to implement the heterogeneous storage connection NAS function and to increase the storage capacity by adding  
15 external storages can realize a large scaled NAS which excels in the capacity scalability and can store a large capacity of file data.

Also, the host can access a file system in an external storage system through a storage system for  
20 controlling input/output operations to/from external storages connected thereto, without the need for being conscious of the file system which has been established in a storage area of the external storage system.

Next, a second embodiment will be described.  
25 The second embodiment applies the heterogeneous storage connection NAS function discussed in connection with the first embodiment.

Fig. 13 illustrates a first exemplary

application. Fig. 13 differs from Fig. 10 in that the CHNO of the storage 1 creates the logical device SLDEV0 on the heterogeneous storage 500, with the file system SFS0 being created on the logical device SLDEV0, and  
5 that a logical device PLDEV0 (180) is created on a disk 17xx of the storage 1, with a file system PFS0 (190) being created on the logical device PLDEV (180).

Another difference lies in that a root directory is mounted at a mount point m1 of the PFS0.  
10 The CHNO has recognized that the root of the SFS0 has been mounted at the mount point m1 of the PFS0. The host 0(400) recognizes the two file systems PFS0 and SFS0 as a single file system PFS0 which is configured in the form of a single directory tree.

15 Assume herein that a method of coupling the SFS0 as part of the PFS0 so that they appear as if they made up a single directory tree is represented herein by "mount." As an example, by soft-linking (symbolic-linking) the root of the SFS0 from the mount point m1  
20 of the PFS0, the SFS0 can be coupled to the PFS0. Another method may create a file system such that the root of the SFS0 is mapped to the mount point m1.

The CHNO can mount the SFS0 at the mount point m1 of the PFS0 by a similar method of mounting  
25 another FS created on another FDEV in the storage 1 in a file system created in a VDEV in the storage 1. This is because the CHNO handles the VDEV in the storage 1 and the VDEV created on the LU in the heterogeneous

storage 500 as the same VDEV, and creates LDEV, LV, FS on the VDEV.

In this way, in the second embodiment, a file system PFSx created on a logical device PLDEVx within the storage 1 is combined with a file system SFSx created on a logical device SLDEVx defined in the heterogeneous storage 500 to build a file system PFSx which comprises a single directory tree that has a capacity too large to be built only with the internal disk capacity. With this configuration, the host can use a large scaled file system with a transparent single view without being conscious of whether a file system resides in the storage 1 or in the heterogeneous storage 500.

Also, the host 0(400) specifies a file system name PFS0 and a file name when it issues an access request for a file stored in the PFS0 or in the SFS0 mounted in the PFS0. Upon receipt of the access request from the host 0(400), the CHN0 confirms whether an associated file is stored in the PFS0 or SFS0, and subsequently accesses data in the file stored in the heterogeneous storage system in a similar way to the first embodiment if the file is stored in the SFS0.

Next, a third embodiment will be described.

Fig. 14 illustrates a second exemplary application of the first embodiment. While Fig. 14 is basically the same as Fig. 13, an increased number of PLDEV's and SLDEV's are provided in the embodiment

illustrated in Fig. 14.

Assuming that the operation of data started in January 2003, PLDEV0, PLDEV1 have been initially established in the storage 1, and file systems PFS0, 5 PFS1 have been created on the respective logical devices. The PFS0 is defined as a home file system, and the PFS1 is mounted at a mount point m1 to create a single view file system PFS0, thereby providing the host 0(400) with a file service.

Fig. 15A shows an exemplary state of the volume management table 131 at the time of January 2003. In the volume management table 131, "LV" indicates a logical volume recognized by the CHNO. "LUN" indicates the number of a LU recognized by the 15 CHNO. "Storage" indicates a storage which stores the LU, where STR0 represents the storage 1, and STR1 represents the heterogeneous storage 500. "LDEV" indicates an LDEV on which the LU has been established in the storage 1. In the third embodiment, assume that 20 every LU is comprised of a single LDEV. Alternatively, a plurality of LDEV's may be coupled to create a single LU. WORM, which is the acronym of Write Once Read Many, indicates an attribute of permitting a write only once and a read many times, where "1" indicates a 25 storage area which has the WORM attribute. "Migration" indicates whether or not an associated logical volume has been migrated, where "1" indicates a migrated storage area. "FS" indicates the name of a file system

established in the LV. "Remarks" indicates supplementary notes, for example, home indicative of a home file system, a date indicating when data was stored in the file system, and the like.

5           For example, the first row in Fig. 15A indicates that LV0 is comprised of LUN0, LUN0 is comprised of PLDEV0 established within the storage 1, and a file system PFS0 is created (this state is represented by LU0/STR0/PLDEV0/PFS0) and is defined as  
10 a home file system. Similarly, the second row in Fig. 15A indicates that LV1 has the configuration represented by LU1/STR0/PLDEV1/PFS1, and stores a file created in January 2003. In the following, this state is represented by a notation  
15 "LV1/LU1/STR0/PLDEV1/PFS1."

Here, the third embodiment shows an example in which each LV stores only files that were created on the same date.

Next, the operation involved in the LDEV  
20 migration will be described. Here, the migration will be described, by way of example, in an operational strategy which defines that at the beginning of next month, files remaining from the previous month are migrated from the storage 1 to a heterogeneous storage,  
25 and are changed to have the WORM attribute. This operational strategy is managed by DLCM 6001 which is a software application stored in a memory of a management terminal 600 installed external to the storage 1. DLCM

is the acronym of Data Life Cycle Manager. As the DLCM 6001 is executed by the management terminal 600, the DLCM 6001 acquires management information, particularly, information in the volume management 5 table 131 from the storage 1, so that the DLCM 6001 can know the configuration of the storage 1.

Assuming herein that it is, for example, on first February, 2003, the DLCM 6001 on the management terminal 600 first issues an execution instruction to 10 the CHN0 of the storage 1 to create PLDEV2 through the management device 18. Upon receipt of this instruction, the CHN0 creates LV1/LUN2/STR0/PLDEV2/PFS2, and starts a file service.

Next, the DLCM 6001 sets the WORM attribute 15 to files created in the previous month, i.e., "January 2003" based on the operational strategy, and determines to migrate these files from the storage 1 to the heterogeneous storage 500.

Specifically, the DLCM 6001 identifies that 20 the files created in "January 2003" are stored in LV1/LUN1/STR0/PLDEV1/PFS1 based on the information in the volume management table 131 shown in Fig. 15A.

Next, the DLCM 6001 issues an execution instruction to the DKC 11 of the storage 1 to set the 25 WORM attribute to the LV1 through the management device 18. Upon receipt of this instruction, a WORM attribute program 111095 of the CHD0 sets the WORM attribute to the PLDEV1 corresponding to the LV1 which has been

instructed to set the WORM attribute from the management device 18, to prohibit new write accesses. While the WORM attribute program 111095 prohibits new writes in the third embodiment, the present invention 5 is not limited to this particular manner in the third embodiment, but the CHNO or CHDO may be provided with a function of prohibiting new write accesses.

Next, the DLCM 6001 determines the SLDEV 1 in the heterogeneous storage 500 as a destination LDEV for 10 the migration. If no LDEV has been previously created, an LDEV creation instruction is issued to the DKC 11 of the storage 1 through the management device 18. As the CHDO receives this instruction, the heterogeneous storage connection control program 11101 creates the 15 SLDEV1.

Next, the DLCM 6001 issues an execution instruction to the DKC 11 of the storage 1 to migrate data in the PLDEV 1 to the SLDEV1 through the management device 18. As the CHDO receives this 20 instruction, the LDEV migration control program 11104 copies data contents in the PLDEV1 to the SLDEV1, i.e., heterogeneous storage 500 through the heterogeneous storage connection SAN 30. Upon completion of the copy, the LDEV migration control program 11104 erases 25 the data in the PLDEV1. Upon completion of the migration, the volume control program 111093 of the CHDO updates the volume management table 131.

Here, unlike new creation of the file system

SFS1, information in the PFS1 has been moved as it is to the SLDEV, so that the CHNO can create the SFS1 only by updating the name of the file system, and updating only other necessary data as required.

5           Thus, the migration has been completed for the LV1 which stores the file system PFS1 created in January. As a result, the volume management table 131 has changed as shown in Fig. 15B. The LV1 is represented as LV1/LUN1/STR1/SLDEV1/SFS1, and is set to  
10 have the WORM attribute.

Subsequently, in a similar manner, at the beginning of March 2003, the DLCM 6001 creates a logical volume LV3 for storing a file system PFS3 for storing files which will be created in March, and  
15 migrates the LV2 which has stored the file system PFS2 in February from the storage 1 to the heterogeneous storage 500. As a result, the LV2 is represented as LV2/LUN2/STR1/SLDEV2/SFS2, as shown in Fig. 15C, and is set to have the WORM attribute. As a result of the  
20 foregoing, the configuration is finally defined as shown in Fig. 14.

Thus, the PLDEV1 and PLDEV2 have been migrated to the SLDEV1 and SLDEV2, respectively, in which case the LV1/LU1 or LV2/LU2 have not at all been  
25 changed. Since the file system after the migration is mounted to the home file system PFS0, the executed migration is not recognized from the host. Also, since the LDEV migration control program manages address

pointers in course of the migration, an access from a host, if any during the migration, would not be interrupted, or would not result in a transfer of erroneous data. As a result, perfect transparency and 5 single view are ensured over the overall file system from the host. Such a function is called the "Single View migration based on heterogeneous storage connection NAS."

In the foregoing description, the creation of 10 a volume and the migration are executed by the DLCM 6001 on the management terminal 600. Alternatively, the function of the DLCM 6001 may be incorporated in an application in the host, or the DLCM 6001 may be run on the host, or the DLCM 6001 may be run on the management 15 device 18 within the storage. Particularly, when the DLCM function is incorporated in an application in the host, more accurate control can be conducted to fit to the characteristics and operation of the application. In this event, an interface between the DLCM 6001 and 20 storage 1 is defined as API, so that the individual operation described above is provided as API.

The foregoing description excludes 25 differences except for those in the configuration of the controller between the storage 1 and heterogeneous storage 500. For example, the storage 1 may employ high performance disks 17xx each having an FC interface, while the heterogeneous storage 500 may employ inexpensive disks 5170x each having an ATA

interface, such that the high performance FC disks may be utilized in a period in which rewrites are frequently performed from the establishment of a file system, while the low-cost AA disks may be utilized in 5 a period in which the storage is a main purpose. The resulting storage system provides high cost performance through such purpose use. This is particularly effective in building a storage system which employs low-cost ATA disks for archiving data over time. The 10 storage system can be adapted to a variety of applications such as a backup of a file system, archiving of E-mails, archiving of logs, archiving of monitoring images, and the like, thus presenting an extremely high utility value.

15         With the foregoing embodiments, independent file systems can be created over a plurality of logical volumes of storages, and combined into a single tree, so that the host can be provided with a single view file system.

20         Also, a logical volume can be created in a storage area on a heterogeneous storage, and an entire file system can be migrated by a migration function of a heterogeneous storage connection function, while maintaining the single view feature.

25         Also, in the event of migration, the WORM attribute can be added to a logical volume to prohibit the logical volume from rewriting.

Further, the creation of a volume and the

migration can be carried out in accordance with a predefined strategy through a control from a management terminal external to the storage or from the host.

It should be further understood by those  
5 skilled in the art that although the foregoing  
description has been made on embodiments of the  
invention, the invention is not limited thereto and  
various changes and modifications may be made without  
departing from the spirit of the invention and the  
10 scope of the appended claims.